

Installing and configuring a MySQL-server on debian

Christian Mertes
mertes@in.tum.de

Dr. Laszlo Kajan
I12 - Department for Bioinformatics and Computational Biology
Technische Universität München

19. Jul 2012

Overview

- Installing the MySQL-packages
- User-Management
- Creating and filling a database
- Basic maintenance work
- Storage Engines

Installing the MySQL-packages

Just install the debian meta-package *mysql-server* with *apt-get*

```
1 apt-get install mysql-server
```

Installing the MySQL-packages

Just install the debian meta-package *mysql-server* with *apt-get*

```
1 apt-get install mysql-server
```

With Debian “Squeeze” we will install:

- MySQL-Server 5.1
- MySQL-Client 5.1
- libmysqlclient16
- MySQL-common
- ...

Installing the MySQL-packages

Just install the debian meta-package *mysql-server* with *apt-get*

```
1 apt-get install mysql-server
```

With Debian “Squeeze” we will install:

- MySQL-Server 5.1
- MySQL-Client 5.1
- libmysqlclient16
- MySQL-common
- ...

The current generally available release of MySQL-server is: 5.5.25

Connecting to MySQL

After installing the meta-package *mysql-server* the MySQL-server is already running.

The server is listening for connections by default under *127.0.0.1* on port *3306*

Connecting to MySQL

After installing the meta-package *mysql-server* the MySQL-server is already running.

The server is listening for connections by default under *127.0.0.1* on port *3306*

How to connect to the MySQL-server in bash:

```
1 mysql -umyUser -pmyPassword \  
2     -hlocalhost -P3306 [db_name]
```

User-Management

MySQL has its own user management.

You can set up user accounts and manage their privileges.

Those information are stored in the *mysql* database.

User-Management

MySQL has its own user management.

You can set up user accounts and manage their privileges.

Those information are stored in the *mysql* database.

Some useful statements to manipulate user accounts:

- CREATE USER
- DROP USER
- GRANT
- RENAME USER
- REVOKE
- SET PASSWORD

Creating an user account

CREATE USER Syntax

```
1 CREATE USER 'user' ['@'host']  
2      [IDENTIFIED BY 'password'];
```

Creating an user account

CREATE USER Syntax

```
1 CREATE USER 'user' ['@'host']  
2     [IDENTIFIED BY 'password'];
```

*Creating user *u1* and *u2**

```
1 CREATE USER 'u1' '@'localhost '  
2     IDENTIFIED BY 'password';  
3  
4 CREATE USER 'u2' '@'%;
```

Managing the privileges of an users

GRANT Syntax

```
1 GRANT privilege ON database TO user ;
```

Managing the privileges of an users

GRANT Syntax

```
1 GRANT privilege ON database TO user ;
```

Apply GRANT on user u1 and u2

```
1 GRANT ALL ON *.* TO 'u1'@'localhost' ;
```

```
2  
3 GRANT SELECT ON *.* TO u2 ;
```

Managing the privileges of an user

remove privileges with REVOKE

```
1 REVOKE grant option ON *.* FROM u1;  
2  
3 REVOKE ALL ON mysql.* FROM 'u2'@'%' ;
```

Managing the privileges of an user

remove privileges with REVOKE

```
1 REVOKE grant option ON *.* FROM u1;  
2  
3 REVOKE ALL ON mysql.* FROM 'u2'@'%';
```

Show privileges of an user

```
1 SHOW GRANTS FOR u1@%;
```

Database-Management

Common workflow of creating a database:

- 1 create database/schema
- 2 create tables
- 3 fill tables with information
- 4 create indices for non primary keys

Database-Management

Common workflow of creating a database:

- 1 create database/schema
- 2 create tables
- 3 fill tables with information
- 4 create indices for non primary keys

To manage your databases MySQL supplies the following statements:

- CREATE DATABASE|TABLE|INDEX
- ALTER DATABASE|TABLE
- DROP DATABASE|TABLE|INDEX
- RENAME TABLE

Creating a database

CREATE DATABASE Syntax

```
1 CREATE {DATABASE|SCHEMA} [IF NOT EXISTS]  
2      db_name [create_specification]
```

Creating a database

CREATE DATABASE Syntax

```
1 CREATE {DATABASE|SCHEMA} [IF NOT EXISTS]  
2      db_name [create_specification]
```

Creating the database *db1*

```
1 CREATE DATABASE db1;
```

Creating a table in a database

```
1 # change to database db1
2 USE db1;
3
4
5 CREATE TABLE myTable (
6     id1 INT(11) AUTO_INCREMENT,
7     id2 INT(11),
8     name VARCHAR(55) NOT NULL,
9     PRIMARY KEY (id1 , id2 ),
10    INDEX (name)
11 ) ENGINE = MYISAM;
```

MySQL-Maintenance

Common things an admin wants to do:

- ❶ do backups
- ❷ read out server status
- ❸ analyze server workload/connections
- ❹ setup a replication of the database

Backing up a database

There are two ways to backup a database:

using a bashtool *mysqldump*

```
1 mysqldump -umyUser -pmyPassword db_name \  
2          > db_name_backup.sql
```

Backing up a database

There are two ways to backup a database:

using a bashtool *mysqldump*

```
1 mysqldump -umyUser -pmyPassword db_name \  
2          > db_name_backup.sql
```

using MySQL built-in methods

```
1 SELECT * INTO OUTFILE 'file_name' FROM myTable;
```

Backing up a database

There are two ways to backup a database:

using a bashtool *mysqldump*

```
1 mysqldump -u myUser -p myPassword db_name \  
2         > db_name_backup.sql
```

using MySQL built-in methods

```
1 SELECT * INTO OUTFILE 'file_name' FROM myTable;
```

You will find the file in */var/lib/mysql/db_name/file_name* on the *localhost*.

Read out server status

As always there many ways to do this

using MySQL built-in methods

```
1 SHOW STATUS [LIKE 'var_name'];
```

Read out server status

As always there many ways to do this

using MySQL built-in methods

```
1 SHOW STATUS [LIKE 'var_name'];
```

using a bashtool *mysqlreport*

```
1 mysqlreport --user=myUser --password=myUser
```

Read out server status

Show running processes/connection on the server

```
1 SHOW FULL PROCESSLIST;
```

Read out server status

Show running processes/connection on the server

```
1 SHOW FULL PROCESSLIST;
```

- *mysqlreport* also gives you the *maximal/current* number of connections.

Read out server status

Show running processes/connection on the server

```
1 SHOW FULL PROCESSLIST;
```

- *mysqlreport* also gives you the *maximal/current* number of connections.
- If you reach the connection limit the mysql-server **refuses any connection!**

Setup a slave for replication

For setting up a slave the main steps are:

- 1 setup a MySQL-Database on a different machine

Setup a slave for replication

For setting up a slave the main steps are:

- 1 setup a MySQL-Database on a different machine
- 2 create the database/schema and tables on the slave.

Setup a slave for replication

For setting up a slave the main steps are:

- 1 setup a MySQL-Database on a different machine
- 2 create the database/schema and tables on the slave.
- 3 let the master listen on the **IP-address** instead of the **localhost**.
in `/etc/mysql/my.conf` set: "*bind-address = 192.168.16.7*"

Setup a slave for replication

For setting up a slave the main steps are:

- 1 setup a MySQL-Database on a different machine
- 2 create the database/schema and tables on the slave.
- 3 let the master listen on the **IP-address** instead of the **localhost**.
in `/etc/mysql/my.conf` set: "***bind-address = 192.168.16.7***"
- 4 add a ***replication-user*** on the master

Setup a slave for replication

For setting up a slave the main steps are:

- 1 setup a MySQL-Database on a different machine
- 2 create the database/schema and tables on the slave.
- 3 let the master listen on the **IP-address** instead of the **localhost**.
in `/etc/mysql/my.conf` set: "*bind-address = 192.168.16.7*"
- 4 add a *replication-user* on the master
- 5 configure master/slave via **my.conf**

Setup a slave for replication

For setting up a slave the main steps are:

- 1 setup a MySQL-Database on a different machine
- 2 create the database/schema and tables on the slave.
- 3 let the master listen on the **IP-address** instead of the **localhost**.
in `/etc/mysql/my.conf` set: "*bind-address = 192.168.16.7*"
- 4 add a *replication-user* on the master
- 5 configure master/slave via **my.conf**
- 6 start replication on slave with **START SLAVE**

Setup a slave for replication

Master-Configuration in my.conf

```
1 [mysqld]
2 ...
3 server-id      = 1
4 log-bin        = /var/log/mysql/mysql-bin.log
5 log-error      = /var/log/mysql/mysql-bin.err
6 # binlog_do_db = db1
7 ...
```

Setup a slave for replication

Slave-Configuration in my.conf

```
1 [mysqld]
2 ...
3 server-id           = 2
4 master-host         = 192.168.16.7
5 master-user          = replicaUser
6 master-password     = myPassword
7 master-connect-retry = 60
8 #replicate --do-db=db1
9 ...
```

Setup a slave for replication

Now you have to initialize the data on the slave.

Load data into slave with mysqldump

```
1 @master$ mysqldump -umyUser -pmyPasswd db_name \  
2 > db_name_backup.sql  
3 @slave$ mysql -umyUser -pmyPasswd db_name \  
4 < db_name_backup.sql
```

Setup a slave for replication

Now you have to initialize the data on the slave.

Load data into slave with mysqldump

```
1 @master$ mysqldump -umyUser -pmyPasswd db_name \  
2 > db_name_backup.sql  
3 @slave$ mysql -umyUser -pmyPasswd db_name \  
4 < db_name_backup.sql
```

Load data into slave with MySQL built-in statements

```
1 @slave$ LOAD DATA FROM MASTER;
```

Storage engines

A storage engine tells MySQL how to handle the data.

MySQL provides a variety of storage engines for different purposes

Storage engines

A storage engine tells MySQL how to handle the data.

MySQL provides a variety of storage engines for different purposes

MyISAM

- very fast
- not transaction safe
- table-level locking
- table-data is stored in only one file

Storage engines

A storage engine tells MySQL how to handle the data.
MySQL provides a variety of storage engines for different purposes

MyISAM

- very fast
- not transaction safe
- table-level locking
- table-data is stored in only one file

InnoDB

- transaction safe
- rollback and crash-recovery capabilities
- row-level locking

Storage engines

Table: Installed storage engines in MySQL 5.1

Engine	Support	Comment
InnoDB	YES	Supports transactions, row-level locking, and foreign keys
MRG_MYISAM	YES	Collection of identical MyISAM tables
BLACKHOLE	YES	/dev/null storage engine (anything just disappears)
CSV	YES	CSV storage engine
MEMORY	YES	Hash based, stored in memory, useful for temporary tables
FEDERATED	NO	Federated MySQL storage engine
ARCHIVE	YES	Archive storage engine
MyISAM	DEF	Default engine as of MySQL 3.23 with great performance

Only *InnoDB* is a transaction-safe storage engine and supports recovery from save-points.

Bibliography

joomla.com. Tweaking MySQL Server. http://www.joomlaperformance.com/articles/server_related/tweaking_mysql_server_23_16.html, June 2012.

Oracle. MySQL Reference Manual 5.1.
<http://dev.mysql.com/doc/refman/5.1/en/>, June 2012.

F. Timme. How To Set Up Database Replication In MySQL 5.1.
http://www.howtoforge.com/mysql_database_replication, June 2012.